



RustCupGame Whitepaper

v 1.0.0



Mission

Our goal is to develop a fully decentralized game and SDK for its growth, taking into account all the current advances in the field of decentralization without a single point of failure.

Modern GameFi projects use blockchain mostly for NFT storage or as a source of in-game currency. We, on the other hand, see more blockchain's potential in gaming:

01

Modern blockchains have high speed and unlimited scalability.

02

Blockchains such as Everscale can be used to build fully decentralized games.



According to existing standards available on the network, it is possible to make truly decentralized metaverses with their own logic and rules.

In this case, the game can be developed not only through the connection of new NFTs or game tokens, but also through game mechanics, adding their own levels and rules. This is similar to the plugin system and extension for existing games.

It will also be possible to manage the game project as a DAO and make decentralized decisions about rule changes and the inclusion of new smart contracts in the game processes



Game mechanics

Game lobby

There is a game lobby organized on the blockchain, where each player can create a bid for the game, wait for opponents, and then the race will automatically start.

Races

The game implements a racing rally with indirect control. To play the game you need to buy racing cars, each of which is an NFT. Before the race the player chooses the parameters for the race: driving style, number of pit stops and others. Once all competitors have entered and set up their NFTs for the race, the algorithm performs an automatic calculation of the winners based on the track parameters. Each race takes place on a unique, automatically generated track.

Connecting custom NFTs as a Drivers

The game implements a mechanism which connects any NFT in the Everscale blockchain as a playable "Driver" object, that has experience. This allows to upgrade any NFT and increase its value on the secondary market.

The betting system

The game implements a betting system. Users who do not have NFTs can bet on the winners of the race. The betting system is available only for TIP3 tokens.

Rating

The game implements a rating system, which is calculated based on a system similar to the Elo Rating in chess.



Tokenomics

The release of the game token will be worked out in detail and described in the 3rd quarter of 2022. All players will receive a token airdrop based on their activity. Currently, the game works only with Ever, the native token of the Everscale blockchain.

The gameplay is divided into 2 parts:

01 Creating or connecting to a lobby to gather participants

02 Race

When creating and connecting to the lobby, the user pays a bid and a Gas Fee for accepting the application. For example, if the bet is 20 EVER, the user needs to pay 21 EVER. 20 EVER goes to the deposit and the remaining 1 EVER goes to the Gas Fee to add the deposit.

The difference between 1 EVER and Gas Fee is returned to the user. If the lobby creator closes it or the participant decides to leave the lobby, the bet will be returned to the owner.



At the end of the race, all bets are summed up and distributed the following way:

In a group race for 4 competitors:

X EVER	10%	60%	30%
The cost per participant	Platform service fee	The Winner receives	The Second place winner receives

In duel mode:

X EVER	10%	90%
The cost per participant	Platform service fee	The winner receives



DAO

Token holders will have an opportunity to affect the game development:

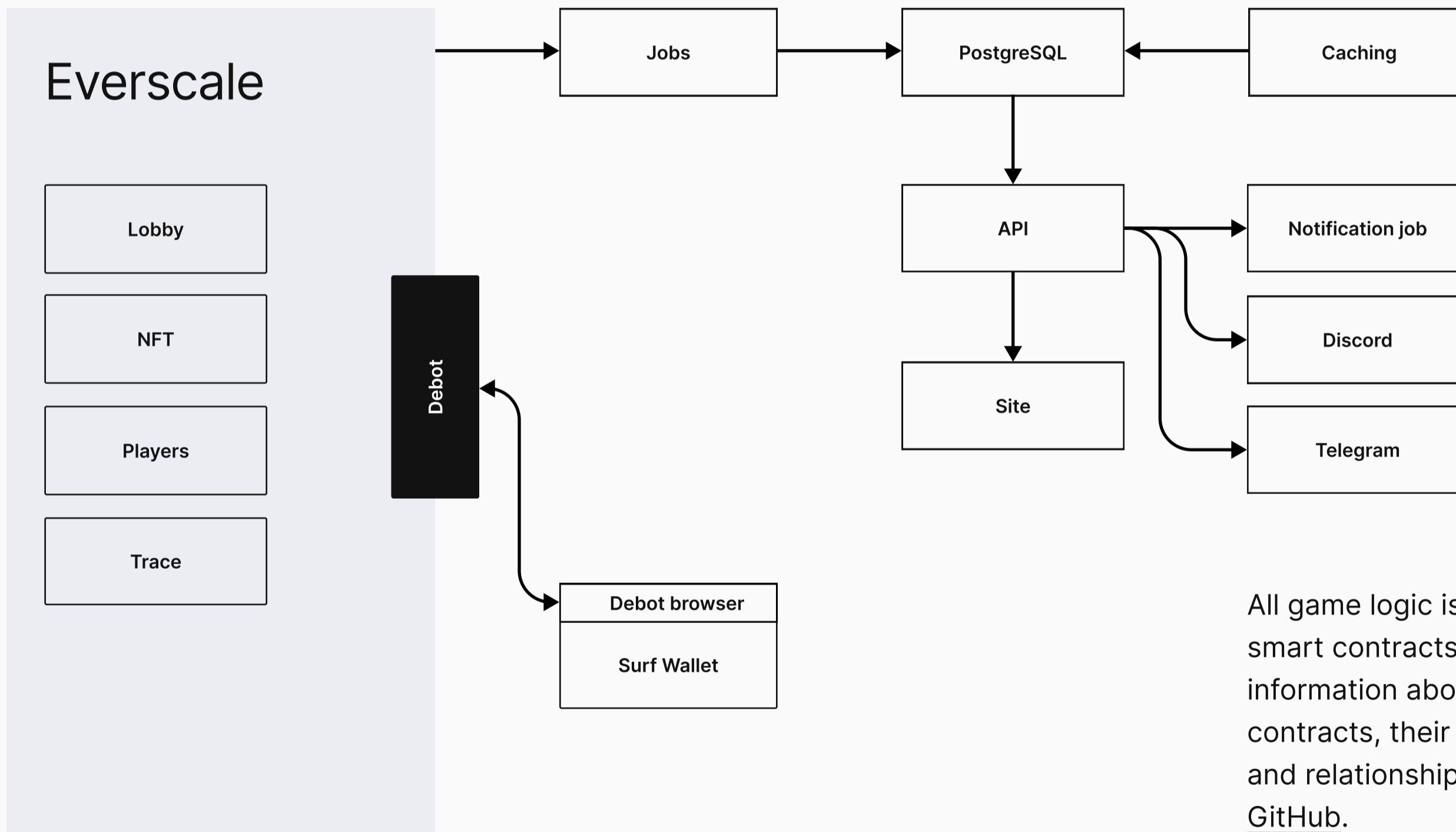
-
- + Participate in the roadmap adoption process

 - + Propose, adopt standards for the plugin system

 - + Vote for the inclusion of game plugins and many other things
-



Technologies and architecture



All game logic is built in a form of the smart contracts system. General information about the used smart contracts, their addresses, interfaces and relationships is published on our [GitHub](#).



REST API tools have been written to display statistics and visualization, which give structured information on the races and players.

The user interaction with the game logic is done through a special smart contract written with DeBot support. For easy access to information about the completed races, all information is collected and indexed in PostgreSQL.

-
- + Telegram-bot

 - + Discord-bot

 - + Website
-

Contacts

[Github](#)

[Telegram](#)

[Discord](#)

[Rustcupgame.com](#)